The Interplay of Population Size and Mutation Probability in the (1+) EA on OneMax

Gießen, Christian; Witt, Carsten

Published in:
Algorithmica

Link to article, DOI:
10.1007/s00453-016-0214-z

Publication date:
2017

Document Version
Peer reviewed version

The Interplay of Population Size and Mutation Probability in the (1+λ) EA on OneMax*

Christian Gießen[1] and Carsten Witt[2]

[1] DTU Compute, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark, cgie@dtu.dk
[2] DTU Compute, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark, cawi@dtu.dk

Abstract

The (1+λ) EA with mutation probability c/n, where $c > 0$ is an arbitrary constant, is studied for the classical OneMax function. Its expected optimization time is analyzed exactly (up to lower order terms) as a function of c and λ. It turns out that $1/n$ is the only optimal mutation probability if $\lambda = o(\ln n \ln\ln n/\ln\ln\ln n)$, which is the cut-off point for linear speed-up. However, if λ is above this cut-off point then the standard mutation probability $1/n$ is no longer the only optimal choice. Instead, the expected number of generations is (up to lower order terms) independent of c, irrespectively of it being less than 1 or greater.

The theoretical results are obtained by a careful study of order statistics of the binomial distribution and variable drift theorems for upper and lower bounds. Experimental supplements shed light on the optimal mutation probability for small problem sizes.

1 Introduction

The runtime analysis of evolutionary algorithms (EAs) is a research area that emerged from the analysis of classical randomized algorithms, where the aim is to prove rigorous statements on the expected runtime and approximation quality of the algorithm depending on the problem size. Since the late 1990s, a number of results on the runtime of simple and moderately complex evolutionary algorithms as well as of other nature-inspired algorithms have emerged [AD11, NW10, Jan13]. Both simple benchmark functions such as OneMax and more complex combinatorial optimization problems were considered. The vast majority of these results are asymptotic, i. e., use O-notation. While such a result in O-notation tells us how the runtime in the worst case scales with

*A preliminary version of this paper was published at GECCO 2015 [GW15].

the problem size, it does not allow a direct comparison of different algorithms or parameter settings in a specific algorithm. For instance, the O-expression omits an implicit constant factor in front of the expression, which might be astronomically large. Hence, an $O(n^3)$-algorithm indeed might be more efficient for practical problem sizes than another $O(n \ln n)$-algorithm. This incomparability even persists if the upper bounds are supplemented by asymptotic lower bounds, e. g., $\Omega(n^3)$. Moreover, if a change of a parameter value such as mutation probability accounts only for a non-asymptotic change of the runtime, this will not become visible in the bound. As a consequence, the optimal setting of the parameter, which minimizes the runtime, is hard to determine.

In recent years, there has been increasing interest in analyses of EAs that are non-asymptotic in the leading term and tight up to lower order terms. Such analyses are more exact than purely asymptotic ones and therefore typically harder to derive. For instance, while the expected runtime of the simple (1+1) EA on OneMax had been known to be $\Theta(n \ln n)$ since the early days of the research area, the first tight lower bound of the kind $(1-o(1))en \ln n$ was not proven until 2010 [DFW10, Sud13]. For the more general case of linear functions, a long series of research results was published (e. g., [Jäg11, DJW12]) until Witt [Wit13] finally proved that the expected runtime of the (1+1) EA equals $(1 \pm o(1))en \ln n$ for any linear function with non-zero weights.

Results like Witt's reveal the leading term in the representation of the expected runtime as a polynomial of n exactly and show that the underlying leading coefficient is in fact small (here $e = 2.71\ldots$). This could not be read off from the classical $\Theta(n \ln n)$ result. However, there is greater potential in such a non-asymptotic analysis. It had for a long time been a rule of thumb to set the mutation probability of the (1+1) EA to $1/n$, i. e., to flip each bit independently with probability $1/n$; however, there was limited theoretical evidence for this. Doerr and Goldberg [DG13] were the first to prove that the $\Theta(n \ln n)$ bound for the (1+1) EA on linear functions also holds if the mutation probability is changed to c/n for an arbitrary positive constant c. Hence, changing the mutation probability to, say, $1/(10n)$ or $10/n$ does not change the asymptotic runtime behavior. Witt's study [Wit13] proves the more general, tight (up to lower order terms) bound $(1 \pm o(1))\frac{e^c}{c} n \ln n$, which exhibits an interesting dependency on the factor $c > 0$ from the mutation probability. Since the factor $\frac{e^c}{c}$ is minimized for $c = 1$, this proves that the most often recommended mutation probability of $1/n$ is optimal for all linear functions.

However, it is by no means clear that $1/n$ is the best choice under all circumstances. For instance, Böttcher, Doerr and Neumann [BDN10] determined the expected optimization time of the (1+1) EA depending on the mutation probability p exactly for the LeadingOnes function. It turned out that the standard choice $p = 1/n$ is not optimal here, but a value of roughly $1.59/n$ minimizes the expected time. A similar result is represented by Sudholt's analysis [Sud12] of a simple crossover-based EA on OneMax, where the optimal mutation probability turns out as $1.618/n$. Note that in both cases a more aggressive mutation than the standard choice is beneficial.

Our study continues this line of research on tight analyses; however, in addition to the mutation probability, we consider a second parameter, namely the *offspring population* size of the underlying evolutionary algorithm. More precisely, we analyze the (1+λ) EA with mutation probability c/n, where c is an arbitrary positive constant, on the classical OneMax function. Our aim is to describe the expected runtime as a function of both the parameter λ (the population size) and the parameter c, in a non-asymptotically tight (up to lower order terms) manner. One aim is to determine the best parameter setting of c, depending on λ. We also pick up the so far quite limited line of work on the (1+λ) EA [JJW05, DK13, DK15], which has been restricted to the standard mutation probability $1/n$, and where OneMax was analyzed before. From this line of work, the bound $\Theta(\frac{n \ln n}{\lambda} + n \frac{\ln \ln \lambda}{\ln \lambda})$ on the expected number of generations has been known; the computational effort in terms of the number of function evaluations is by a factor of λ bigger. A remarkable insight drawn from this bound is that the number of generations enjoys a linear speed-up with respect to λ as long as $\lambda = o(\ln n \ln \ln n / \ln \ln \ln n)$. If λ is above this threshold (the so-called *cut-off* point), the second term from the time bound becomes relevant and the linear speed-up ceases to exist.

Note that in the following we will use the term population size synonymously with offspring population size. The concept of parent populations has been studied previously from a theoretical perspective [Wit06]. However, to the best of our knowledge, no asymptotically tight analyses have been performed for parent populations.

Another insight already known from previous work is that choosing $\lambda = 1$ yields at least asymptotically the best choice to minimize the expected number of f-evaluations; with respect to the standard mutation probability, this has actually been proven to be the absolute truth without asymptotics [JJW05]: increasing λ never decreases the expected number of f-evaluations. Therefore, we are primarily interested in optimizing the factor c in the mutation probability c/n depending on λ, assuming a sufficiently large parallel architecture allowing the λ offspring evaluations to be done in parallel. Note also that our work and the work we relate it to assumes a static choice of λ, fixed throughout the whole run of the (1+λ) EA. Using adaptive offspring population sizes depending on the current fitness value, an asymptotically different runtime behavior may occur [BLS14]. Also a static choice of c is assumed and dynamic schedules for the mutation probability, as studied for a (1+1) EA in [BDN10], are not within the scope of this research.

In this paper, we prove that the expected runtime (i. e., number of generations) of the (1+λ) EA with mutation probability c/n on OneMax equals

$$(1 \pm o(1)) \left(\frac{e^c}{c} \cdot \frac{n \ln n}{\lambda} + \frac{1}{2} \cdot \frac{n \ln \ln \lambda}{\ln \lambda} \right),$$

which greatly generalizes the previous $\Theta(\frac{n \ln n}{\lambda} + n \frac{\ln \ln \lambda}{\ln \lambda})$ bound. Hence, as long as λ is below the cut-off point, more precisely, if $\lambda = o(\ln n \ln \ln n / \ln \ln \ln n)$, the leading term of the expected runtime is the same as for linear functions with the

(1+1) EA, and setting $c = 1$ minimizes the expected runtime. However, when λ is above the cut-off, more precisely, if $\lambda = \omega(\ln n \ln\ln n / \ln\ln\ln n)$, such that the second term becomes the leading term, one may choose c as an arbitrarily small or large constant without changing the expected runtime, up to lower order terms. So somewhat counter-intuitively, mutations are also allowed to occur less frequently than in a (1+1) EA. Any effect on the lower order term of the expected runtime is not explained by this result, though. These exact results follow from a careful study of order statistics of the binomial distribution and are obtained by variable drift theorems for upper and lower bounds.

Altogether, our study gives advice on the choice of the mutation probability and also determines the cut-off point for speed-up for different mutation probabilities. To the best of our knowledge, this represents the first tight (up to lower order terms) runtime analysis of a population-based EA, and it seems also be novel in developing such a tight expression for the runtime depending on two parameters. We remark that our analyses build on the insights by Doerr and Künnemann [DK15], who also give asymptotic results holding for arbitrary linear functions.

This paper is structured as follows. Section 2 introduces the framework, and presents drift theorems as well as properties of order statistics used for the analysis. Section 3 proves our main result and discusses its implications. Supplemental experimental studies are presented in Section 4. We finish with some conclusions.

2 Preliminaries

2.1 Algorithm

We consider the (1+λ) EA for the minimization of pseudo-boolean functions $f\colon \{0,1\}^n \to \mathbb{R}$, defined as Algorithm 1. The case of $c = 1$ in the mutation probability was considered in [JJW05, DK13, DK15]. If both $\lambda = 1$ and $c = 1$, the algorithm simplifies to the classical (1+1) EA [AD11]. Throughout the paper, c is assumed to be constant, i. e., it may not depend on n.

Algorithm 1 (1+λ) EA

```
Select x* uniformly at random from {0,1}^n.
for t ← 1, 2, ... do
    for i ← 1, ..., λ do
        Create x_i by flipping each bit of x* independently with probability c/n.
    x_m ← arg min_{x_i} f(x_i) (breaking ties randomly)
    if f(x_m) ≤ f(x*) then
        x* ← x_m
```

The *runtime*, also called the *optimization time*, of the (1+λ) EA is the smallest t such that an individual of minimum f-value has been found. Note that t corresponds to a number of iterations (also called generations), where each

generation creates λ offspring. Since each of these offspring has to be evaluated, the number of function evaluations, which is a classical cost measure, is by a factor of λ larger than the runtime as defined here. However, assuming a massively parallel architecture that allows for parallel evaluation of the offspring, counting the number of generations seems also a valid cost measure. In particular, a speed-up on the function $\textsc{OneMax}(x_1, \dots, x_n) := x_1 + \cdots + x_n$ by increasing λ can only be observed in terms of the number of generations. As proven by Jansen, De Jong and Wegener [JJW05], the (1+1) EA performs on OneMax stochastically at most the same number of function evaluations as any (1+λ) EA for $\lambda > 1$ and $c = 1$. For arbitrary c and even more general population-based algorithms, a corresponding statement is proven in [Wit13]. Note that for reasons of symmetry, it makes no difference whether OneMax is minimized (as in the present paper) or maximized (as in several previous research papers).

Throughout the paper, all O-notation (mostly of the kind "$o(1)$") will be with respect to the problem size n.

2.2 Drift Theorems

Our results are obtained by variable drift analysis, which is also used in the asymptotic analysis of the (1+λ) EA on OneMax and other linear functions [DK15]. The first theorems stating upper bounds on the hitting time using variable drift go back to [Joh10, MRC09]. These theorems were subsequently generalized in [RS14] and [LW14]. We use the latter version.

Theorem 1 (Variable Drift, Upper Bound; [LW14])**.** *Let $(X_t)_{t\geq 0}$, be a stochastic process adapted to a filtration $\mathcal{F}_t$ over some state space $S \subseteq \{0\} \cup [x_{\min}, x_{\max}]$, where $x_{\min} > 0$. Let $h(x) \colon [x_{\min}, x_{\max}] \to \mathbb{R}^+$ be a monotone increasing function such that $1/h(x)$ is integrable on $[x_{\min}, x_{\max}]$ and $E(X_t - X_{t+1} \mid \mathcal{F}_t) \geq h(X_t)$ if $X_t \geq x_{\min}$. Then it holds w. r. t. the first hitting time $T := \min\{t \mid X_t = 0\}$ that (assuming the following expectation to exist)*

$$E(T \mid X_0) \leq \frac{x_{\min}}{h(x_{\min})} + \int_{x_{\min}}^{X_0} \frac{1}{h(x)} \, \mathrm{d}x \ .$$

To prove lower bounds on the hitting time by variable drift, we need additional assumptions like the one in the following lemma, a special case of which was first proposed in [DFW11].

Theorem 2 (Variable Drift, Lower Bound; [LW14])**.** *Let $(X_t)_{t\geq 0}$, be a stochastic process adapted to a filtration $\mathcal{F}_t$ over some state space $S \subseteq \{0\} \cup [x_{\min}, x_{\max}]$, where $x_{\min} > 0$. Suppose there exists two functions $\xi, h \colon [x_{\min}, x_{\max}] \to \mathbb{R}^+$ such that $h(x)$ is monotone increasing and $1/h(x)$ integrable on $[x_{\min}, x_{\max}]$, and for all $t \geq 0$,*

(i) $X_{t+1} \leq X_t$,

(ii) $X_{t+1} \geq \xi(X_t)$ *for* $X_t \geq x_{\min}$,

(iii) $E(X_t - X_{t+1} \mid \mathcal{F}_t) \leq h(\xi(X_t))$ *for* $X_t \geq x_{\min}$.

Then it holds for the first hitting time $T := \min\{t \mid X_t = 0\}$ *that*

$$E(T \mid X_0) \geq \frac{x_{\min}}{h(x_{\min})} + \int_{x_{\min}}^{X_0} \frac{1}{h(x)} \, \mathrm{d}x \ .$$

2.3 Tight Bounds on Order Statistics

When the (1+λ) EA optimizes OneMax, it samples λ offspring and takes the so-called winner individual from the offspring having minimum OneMax-value, breaking ties arbitrarily. This problem is strongly related to analyzing how many one-bits are flipped in the offspring that flips most one-bits (however, this is not necessarily the winner individual). The number of flipping one-bits per offspring follows a binomial distribution, which is why the following results on order statistics of the binomial distribution are crucial. We do not claim these statements on order statistics to be fundamentally new; however, we did not find them in the literature in this form. In the following we use the notation $X \sim Y$ for two random variables X and Y to denote that X follows the same distribution as Y.

We start with an auxiliary lemma that will be useful in the course of this section. It is a well-known inequality, often used in the analysis of evolutionary algorithms. For completeness, we give a formal proof.

Lemma 3. *Let* $X \sim \mathrm{Bin}(n,p)$. *Then,* $\Pr(X \geq k) \leq \binom{n}{k} p^k$ *for all* $k \geq 0$.

Proof. By definition, the probability of the event $X \geq k$ is

$$\Pr(X \geq k) = \sum_{j=0}^{n-k} \binom{n}{k+j} p^{k+j} (1-p)^{n-k-j}.$$

Using

$$\binom{n}{k+j} = \frac{n!(n-k)!}{k!(k+1)\cdots(k+j)(n-k-j)!(n-k)!} \leq \binom{n}{k}\binom{n-k}{j},$$

which follows from $k \geq 0$, we get

$$\begin{aligned}\Pr(X \geq k) &\leq \sum_{j=0}^{n-k} \binom{n}{k}\binom{n-k}{j} p^{k+j}(1-p)^{n-k-j} \\ &= \binom{n}{k} p^k \sum_{j=0}^{n-k} \binom{n-k}{j} p^j (1-p)^{n-k-j} = \binom{n}{k} p^k,\end{aligned}$$

where the last step is using the binomial identity. □

We can now state our main result on order statistics.

Lemma 4. *Let X_i, where $i \in \{1, \ldots, \lambda\}$, independent random variables being identically distributed as $X_i \sim \mathrm{Bin}(k, c/n)$ for some $k \leq n$ and constant $c > 0$. Let $X^* := \max\{X_1, \ldots, X_\lambda\}$ be the maximum order statistic of the X_i. Then*

1. *If $\lambda ck/n = o(1)$ then $E(X^*) = (1 - o(1))\lambda E(X_1) = (1 - o(1))\lambda ck/n$.*
2. *If $\lambda ck/n \geq \alpha$ for $\alpha > 0$ then $E(X^*) \geq \alpha/(1 + \alpha)$.*
3. *If $\lambda = \omega(1)$ and $k = n/(\ln^\alpha \lambda)$, where $\alpha \geq 0$ and $\alpha = O(1)$, then $E(X^*) = \frac{(1 \pm o(1))}{1+\alpha} \frac{\ln \lambda}{\ln \ln \lambda}$.*

Proof. We start with the first item. Note that for $j \geq 0$,

$$\Pr(X^* = j) \geq \binom{\lambda}{1} \Pr(X_1 = j) \left(\Pr(X_1 < j)\right)^{\lambda - 1} \quad ,$$

since the maximum equals j if exactly one of the variables takes this value and the remaining ones are less; to denote a single of the identically distributed random variables, we always refer to X_1.

The last bound is

$$\lambda \Pr(X_1 = j) \left(1 - \Pr(X_1 \geq j)\right)^{\lambda - 1} \geq \lambda \Pr(X_1 = j) \left(1 - \lambda \Pr(X_1 \geq j)\right)$$

according to Bernoulli's inequality. If $j \geq 1$ then

$$\Pr(X_1 \geq j) \leq E(X_1)/j \leq E(X_1)$$

by Markov's inequality. Altogether,

$$\begin{aligned} \Pr(X^* = j) &\geq \lambda \Pr(X_1 = j)(1 - \lambda E(X_1)) \\ &= (1 - o(1))\lambda \Pr(X_1 = j) \end{aligned}$$

by our assumption that $\lambda E(X_1) = \lambda ck/n = o(1)$. The item now follows since

$$\begin{aligned} E(X^*) &= \sum_{j=1}^{n} j \cdot \Pr(X^* = j) \\ &\geq \sum_{j=1}^{n} j(1 - o(1))\lambda \Pr(X_1 = j) \\ &= (1 - o(1))\lambda E(X_1) \end{aligned}$$

along with the trivial upper bound $E(X^*) \leq \lambda E(X_1)$.

To show the second item, we note that

$$\begin{aligned} E(X^*) &= \sum_{\ell \geq 1} \Pr(X^* \geq \ell) \\ &\geq \Pr(X^* \geq 1) \\ &= 1 - \left(\Pr(X_1 = 0)\right)^\lambda \quad , \end{aligned}$$

since the maximum is at least 1 if it does not happen that all λ random variables evaluate to 0. Hence, using $e^x \geq 1+x$ and $e^{-x} \leq 1/(1+x)$ for $x \in \mathbb{R}$, we have

$$\begin{aligned}
E(X^*) &\geq 1 - \left(\left(1-\frac{c}{n}\right)^k\right)^\lambda \\
&\geq 1 - \left(\left(1-\frac{c}{n}\right)^{\frac{\alpha n}{c\lambda}}\right)^\lambda \\
&\geq 1 - \left(e^{-\frac{\alpha}{\lambda}}\right)^\lambda \\
&= 1 - e^{-\alpha} \\
&\geq 1 - \left(\frac{1}{1+\alpha}\right) \\
&= \frac{\alpha}{1+\alpha} \ .
\end{aligned}$$

To prove the third item, we will first show $E(X^*) \leq (1+o(1))\frac{\ln\lambda}{(1+\alpha)\ln\ln\lambda}$ and then $E(X^*) \geq (1-o(1))\frac{\ln\lambda}{(1+\alpha)\ln\ln\lambda}$. We use that

$$\begin{aligned}
\Pr(X^* \geq j) &= 1 - (\Pr(X_1 < j))^\lambda \\
&= 1 - (1 - \Pr(X_1 \geq j))^\lambda \ .
\end{aligned}$$

Furthermore, by using Lemma 3 we get

$$\begin{aligned}
\Pr(X_1 \geq j) &\leq \binom{n/(\ln^\alpha \lambda)}{j} \left(\frac{c}{n}\right)^j \\
&\leq \left(\frac{c}{\ln^\alpha \lambda}\right)^j \frac{1}{j!} \\
&\leq \left(\frac{ce}{j\ln^\alpha \lambda}\right)^j \ ,
\end{aligned}$$

where the last inequality uses $j! \geq (j/e)^j$. Plugging this in our expression for $\Pr(X^* \geq j)$, we have

$$\Pr(X^* \geq j) \leq 1 - \left(1 - \left(\frac{ce}{j\ln^\alpha \lambda}\right)^j\right)^\lambda \ .$$

The aim now is to inspect the last bound for $j = \frac{\ln\lambda}{(1+\alpha)\ln\ln\lambda} + \beta$, where $\beta \geq \delta(\lambda) := 4\frac{\ln\lambda(\ln\ln\ln\lambda + 1 + \ln c + \ln(1+\alpha))}{(\ln\ln\lambda)^2}$ (the choice of $\delta = \delta(\lambda)$ is to some extent arbitrary). In the following, we will implicitly assume that $\beta \geq 0$, which holds for large enough λ. We will estimate the inner term $(ce/(j\ln^\alpha\lambda))^j = e^{-j(\ln(j/(ce)) + \alpha\ln\ln\lambda)}$ from above. We start with the first term in the last expo-

nent and compute

$$\begin{aligned}
&j(\ln(j/(ce)) \\
&= \left(\frac{\ln\lambda}{(1+\alpha)\ln\ln\lambda} + \beta\right) \ln\left(\frac{\ln\lambda}{(1+\alpha)ce\ln\ln\lambda} + \frac{\beta}{ce}\right) \\
&\geq \left(\frac{\ln\lambda}{(1+\alpha)\ln\ln\lambda} + \beta\right) (\ln\ln\lambda - \ln\ln\ln\lambda - \ln(ce(1+\alpha))) \\
&\geq \frac{\ln\lambda}{1+\alpha} + \frac{\beta(\ln\ln\lambda)^2/2 - \ln\lambda(\ln\ln\ln\lambda + \ln(ce(1+\alpha))}{\ln\ln\lambda} \ ,
\end{aligned}$$

where the last inequality assumes $\ln\ln\lambda - \ln\ln\ln\lambda - \ln(ce(1+\alpha)) \geq (\ln\ln\lambda)/2$, which holds for large enough λ, and uses $1+\alpha \geq 1$.

By definition of δ, we have $\beta(\ln\ln\lambda)^2/2 - \ln\lambda(\ln\ln\ln\lambda + 1 + \ln c + \ln(1+\alpha)) \geq \beta(\ln\ln\lambda)^2/4$, and therefore

$$j\ln(j/(ce)) \geq \frac{\ln\lambda}{1+\alpha} + \frac{\beta}{4}(\ln\ln\lambda) \ .$$

Therefore, the whole exponent is bounded from below according to

$$\begin{aligned}
j(\ln(j/(ce)) + (\alpha\ln\ln\lambda)) &\geq \frac{\ln\lambda}{1+\alpha} + \frac{\beta}{4}(\ln\ln\lambda) \\
&\quad + \alpha(\ln\ln\lambda)\left(\frac{\ln\lambda}{(1+\alpha)\ln\ln\lambda} + \beta\right) \\
&\geq \ln\lambda + \frac{\beta}{4}(\ln\ln\lambda) \ .
\end{aligned}$$

Plugging this in our expression for $\Pr(X^* \geq j)$, we have for $\beta' := \beta/4$

$$\begin{aligned}
\Pr(X^* \geq j) &\leq 1 - \left(1 - e^{-\ln\lambda - \beta'(\ln\ln\lambda)}\right)^{\lambda} \\
&= 1 - \left(1 - \frac{1}{\lambda}e^{-\beta'(\ln\ln\lambda)}\right)^{\frac{\lambda e^{\beta'(\ln\ln\lambda)}}{2} \cdot \frac{2}{e^{\beta'(\ln\ln\lambda)}}} \\
&\leq 1 - e^{-2e^{-\beta'(\ln\ln\lambda)}} \\
&\leq 1 - (1 - 2e^{-\beta'(\ln\ln\lambda)}) = 2e^{-\beta'(\ln\ln\lambda)} \ ,
\end{aligned}$$

using first $(1-1/x)^{x/2} > e^{-1}$ for $x \geq 2$ and then $e^{-x} \geq 1 - x$ for $x \in \mathbb{R}$.

Now,

$$\begin{aligned}
E(X^*) &= \sum_{j\geq 1} \Pr(X^* \geq j) \\
&\leq \frac{\ln\lambda}{(1+\alpha)\ln\ln\lambda} + \delta + \sum_{j \geq \frac{\ln\lambda}{(1+\alpha)\ln\ln\lambda} + \delta} \Pr(X^* \geq j) \ .
\end{aligned}$$

The last sum is now split in sub-sums consisting of δ terms each, where still $\delta = 4\frac{\ln\lambda(\ln\ln\ln\lambda+1+\ln c+\ln(1+\alpha))}{(\ln\ln\lambda)^2}$. We get

$$\begin{aligned}\sum_{j\geq\frac{\ln\lambda}{(1+\alpha)\ln\ln\lambda}+\delta} &\Pr(X^*\geq j)\\ &\leq\sum_{i=1}^{\infty}\delta\cdot\Pr\left(X^*\geq\frac{\ln\lambda}{(1+\alpha)\ln\ln\lambda}+i\delta\right)\\ &\leq\sum_{i=1}^{\infty}2\delta e^{-i(\delta/4)(\ln\ln\lambda)}=o(1)\end{aligned}$$

according to our bound on $\Pr(X^*\geq j)$ and the geometric series. Altogether,

$$E(X^*)\leq\frac{\ln\lambda}{(1+\alpha)\ln\ln\lambda}+\delta+o(1)=\frac{1+o(1)}{1+\alpha}\frac{\ln\lambda}{\ln\ln\lambda}\ ,$$

where the last step used that $\delta=O((\ln\lambda\ln\ln\ln\lambda)/(\ln\ln\lambda)^2)=o(\ln\lambda/(\ln\ln\lambda))$ for $\alpha=O(1)$. This proves the upper bound on $E(X^*)$ from the third item.

We are left with the lower bound. Note that for $j\geq 1$,

$$\begin{aligned}\Pr(X_1\geq j)&\geq\binom{n/\ln^\alpha\lambda}{j}\left(\frac{c}{n}\right)^j\left(1-\frac{c}{n}\right)^{n-j}\\ &\geq\left(\frac{n}{j\ln^\alpha\lambda}\right)^j\left(\frac{c}{n}\right)^j e^{-2c}\\ &=e^{-2c}\left(\frac{c}{j\ln^\alpha\lambda}\right)^j,\end{aligned}$$

where the second inequality used that $(1-c/n)^{n-j}\geq(1-c/n)^n=(1-c/n)^{(n/c)c}\geq e^{-2c}$ for n large enough since $(1-x)^x\geq e^{-2}$ for sufficiently large x. Thus, we get

$$\Pr(X^*<j)\leq\left(1-e^{-2c}\left(\frac{c}{j\ln^\alpha\lambda}\right)^j\right)^\lambda.$$

Similarly to the upper bound, we consider the last bound for $j=\frac{\ln\lambda}{(1+\alpha)\ln\ln\lambda}-\beta$, where $\beta=2\frac{\ln\lambda\ln(c(1+\alpha)\ln\ln\lambda)}{(\ln\ln\lambda)^2}$. We work with the representation $(c/(j\ln^\alpha\lambda))^j=e^{-j(\ln(j/c)+\alpha\ln\ln\lambda)}$. To bound the last exponent, we note that

$$j\ln(j/c)\leq\frac{\ln\lambda}{1+\alpha}-\beta\ln\ln\lambda$$

for λ large enough. Hence, $j\ln(j/c)+j\alpha\ln\ln\lambda\leq\ln\lambda-\beta\ln\ln\lambda$. We have in

total

$$\begin{aligned}\Pr(X^* < j) &\leq \left(1 - e^{-2c}\left(\frac{c}{j \ln^\alpha \lambda}\right)^j\right)^\lambda \\ &\leq \left(1 - e^{-\ln\lambda - 2c + \beta \ln\ln\lambda}\right)^\lambda \\ &\leq \left(1 - \frac{1}{\lambda e^{2c-\beta\ln\ln\lambda}}\right)^{\lambda e^{2c-\beta\ln\ln\lambda}\cdot\frac{1}{e^{2c-\beta\ln\ln\lambda}}} \\ &\leq e^{-e^{-2c+\beta\ln\ln\lambda}} = o(1),\end{aligned}$$

where the second inequality used our bound on $j(\ln(j/c) + \alpha \ln\ln\lambda)$. By the law of total probability, we have that for all j

$$\begin{aligned}E(X^*) &\geq \Pr(X^* \geq j) E(X^* \mid X^* \geq j) \\ &\geq j(1 - \Pr(X^* < j)) \ ,\end{aligned}$$

Plugging in $j = \frac{\ln\lambda}{(1+\alpha)\ln\ln\lambda} - \beta$ immediately gives the desired bound

$$\begin{aligned}E(X^*) &\geq \left(\frac{\ln\lambda}{(1+\alpha)\ln\ln\lambda} - \beta\right)(1 - o(1)) \\ &= \frac{1-o(1)}{1+\alpha}\frac{\ln\lambda}{\ln\ln\lambda} \ .\end{aligned}$$

□

The parameter k used in Lemma 4 above will correspond to the number of one-bits in the current individual (recall that the aim is to create an individual with all zeros since we are minimizing). Hence, the progress can be as big as X^*, the maximum number of flipping one-bits in the λ trials creating offspring. However, it is not guaranteed that the offspring flipping most one-bits is accepted. For instance, it could flip even more zero-bits. Therefore, the following lemma will be used to estimate the probability of accepting the individual related to the number X^*.

Lemma 5. *Assume that an individual with k one-bits is mutated by flipping each bit independently with probability c/n, where $c > 0$ is a constant. The probability that no zero-bit flips in this mutation is at least*

$$(1 - o(1))e^{-c + \frac{kc}{n}} \ .$$

Moreover, the probability that the mutation creates an individual with at most k one-bits (i. e., at most the same OneMax*-value) is at most*

$$(1 + o(1))e^{-c + \frac{k(c+c^2)}{n}} \ .$$

Proof. We start with the first claim. The probability of not flipping a zero-bit equals

$$\left(1 - \frac{c}{n}\right)^{n-k} = \left(1 - \frac{c}{n}\right)^n \left(1 - \frac{c}{n}\right)^{-k} \geq (1 - o(1))e^{-c}e^{kc/n} \ ,$$

where we have used that c is a constant.

To create an individual with at most k one-bits, it is necessary that at least the same number of one-bits as zero-bits flips. The probability of flipping at least j one-bits is bounded from above $\binom{k}{j}\left(\frac{c}{n}\right)^j$ and the probability of flipping exactly j zero-bits equals $\binom{n-k}{j}\left(\frac{c}{n}\right)^j\left(1-\frac{c}{n}\right)^{n-k-j}$. Hence, the probability of obtaining at most k one-bits is at most

$$\begin{aligned}
&\sum_{j=0}^{\max\{k,n-k\}} \binom{k}{j}\binom{n-k}{j}\left(\frac{c}{n}\right)^{2j}\left(1-\frac{c}{n}\right)^{n-k-j} \\
&\leq \left(1-\frac{c}{n}\right)^{n-k}\left(\sum_{j=0}^{\ln n}\binom{k}{j}\binom{n-k}{j}\left(\frac{c}{n}\right)^{2j}\left(1-\frac{c}{n}\right)^{-\ln n}\right. \\
&\qquad\qquad \left. +e^{-\Omega(\ln n \ln\ln n)}\right) \ ,
\end{aligned}$$

where we used that the probability of flipping at least $\ln n$ bits is at most

$$\binom{n}{\ln n}\left(\frac{c}{n}\right)^{\ln n} \leq \frac{c^{\ln n}}{(\ln n)!} = e^{-\Omega(\ln n \ln\ln n)}$$

for any mutation probability c/n. Using $\binom{a}{b} \leq \frac{a^b}{b!}$, $(1-c/n)^{-\ln(n)} = 1+o(1)$ and $\left(1-\frac{c}{n}\right)^{n-k} \leq e^{-c+kc/n}$ and subsuming lower order terms into a $o(1)$-term, our bound is at most

$$\begin{aligned}
&(1+o(1))e^{-c+kc/n}\left(\sum_{j=0}^{\ln n}\frac{k^j(n-k)^j}{j!\cdot j!}\left(\frac{c^2}{n^2}\right)^j\right) \\
&\leq (1+o(1))e^{-c+kc/n}\left(\sum_{j=0}^{\infty}\frac{(c^2(k/n)((n-k)/n))^j}{j!}\right) \\
&= (1+o(1))e^{-c+kc/n}e^{c^2\frac{k}{n}\frac{n-k}{n}} \leq (1+o(1))e^{-c+\frac{k}{n}(c+c^2)} \ ,
\end{aligned}$$

where the last inequality used $(n-k)/n \leq 1$. □

Finally, to a similar purpose as described before Lemma 5, we will need the following lemma to correct the progress made by the individual flipping most one-bits by the number of flipping zero-bits. The statement is given in a general framework without appealing to a particular distribution; the notation $\succ$ denotes stochastic dominance.

Lemma 6. *Let $X_1,\dots,X_\lambda$ and $Y_1,\dots,Y_\lambda$ be two sequences of independent, identically distributed random variables. Let $Z_i := X_i - Y_i$ for $i \in \{1,\dots,\lambda\}$ and $Z^* := \max\{Z_i \mid i \in \{1,\dots,\lambda\}\}$. Then*

$$Z^* \succ \max\{X_i \mid i \in \{1,\dots,\lambda\}\} - Y_1 \ ,$$

where the index 1 without loss of generality refers to any of the Y_i.

Proof. We show the lemma only for $\lambda = 2$; the general case follows in the same way. We abbreviate $\tilde{Z}^* := \max\{X_i \mid i \in \{1, \dots, \lambda\}\} - Y_1$ and note that we have to show

$$\Pr(Z^* \geq x) \geq \Pr\Big(\tilde{Z}^* \geq x\Big)$$

for all $x \in \mathbb{R}$. By definition and using independence

$$\Pr(Z^* \geq x) = 1 - (\Pr(X_1 - Y_1 < x) \cdot \Pr(X_2 - Y_2 < x))$$

and because of dependence

$$\Pr\Big(\tilde{Z}^* \geq x\Big) = 1 - \big(\Pr(X_1 - Y_1 < x) \cdot \Pr(X_2 - Y_1 < x \mid X_1 - Y_1 < x)\big) \ .$$

The lemma follows if we can show that

$$\Pr(X_2 - Y_2 < x) \leq \Pr(X_2 - Y_1 < x \mid X_1 - Y_1 < x).$$

To see this, note that $Y_1 \sim Y_2$ without any conditioning. The condition is $Y_1 > X_1 - x$, so given this lower bound the probability of also $Y_1 > X_2 - x$ happening only becomes larger. □

3 Analysis of (1+λ) EA on OneMax

The following theorem is our main result.

Theorem 7. *Suppose $\lambda \leq n^r$ for some natural number r and let T denote the optimization time (i. e., number of generations) of the (1+λ) EA with mutation probability c/n, where $c > 0$ is a constant, on* OneMax. *Then*

$$E(T) = (1 \pm o(1)) \left(\frac{e^c}{c} \cdot \frac{n \ln n}{\lambda} + \frac{1}{2} \cdot \frac{n \ln \ln \lambda}{\ln \lambda} \right).$$

Before we prove the theorem, we discuss its implications. If we choose $\lambda = o(\ln n \ln \ln n / \ln \ln \ln n)$, the first term is asymptotically largest and we observe the linear speed-up by a factor of λ. Since $\frac{e^c}{c}$ is minimized for $c = 1$, the optimal mutation probability (minimizing the expected optimization time) is $1/n$. Neither mutation probabilities that are by a constant factor larger or by a constant factor smaller are optimal.

If $\lambda = \omega(\ln n \ln \ln n / \ln \ln \ln n)$, i. e., is above the cut-off point for speed-up, the second term from the bound becomes the leading term and it turns out that choosing c arbitrarily small or large (but constant) is asymptotically no worse than the standard choice $c = 1$. A large population makes the algorithm more robust with respect to the choice of the mutation probability.

Proof. Proof of Theorem 7 Let X_t denote the number of one-bits of the current search point of the (1+λ) EA at generation t and $\Delta_t := X_t - X_{t+1}$. Recall that the aim is to reach an X_t-value of 0.

Upper Bound. In order to show the upper bound we will work out a lower bound on the drift. To this end, in some regions only take into account steps of the process where progress towards the optimum occurs from one-bits only, i. e., for some regions we condition the process on the event that no zero-bit flips in the offspring with largest number of flipping one-bits. According to Lemma 5, this probability is at least $(1-o(1))e^{-c+cX_t/n} \geq (1-o(1))e^{-c}$. Conditioning on this event, which considers the mutation of the zero-bits and is independent of the mutation of the one-bits, we can estimate the drift for different regimes w. r. t. X_t.

We first consider the case of $\lambda = \omega(1)$. We have that $E(\Delta_t \mid X_t) \geq h(X_t)$, where

$$h(X_t) := \begin{cases} (1-o(1))\frac{\ln \lambda}{\ln \ln \lambda} & \text{if } X_t \geq \frac{n}{(\ln \lambda)^{\frac{1}{\ln \ln \ln \lambda}}} \\ (1/2-o(1))e^{-c}\frac{\ln \lambda}{\ln \ln \lambda} & \text{if } X_t \geq \frac{n}{\ln \lambda} \\ (1-o(1))e^{-c}\min\{c,1\}/2 & \text{if } X_t \geq \frac{n}{\lambda} \\ (1-o(1))e^{-c}\frac{c}{\sqrt{\ln n}} & \text{if } X_t \geq \frac{n}{\lambda\sqrt{\ln n}} \\ (1-o(1))ce^{-c}\lambda X_t/n & \text{if } X_t < \frac{n}{\lambda\sqrt{\ln n}} \ . \end{cases}$$

The bound on the drift for the first regime can be obtained from $E(\Delta_t \mid X_t) \geq (1-o(1))E(X^*)$, where X^* denotes the maximum number of flipping ones over λ offspring at generation t which can be shown as follows. Let Y_i, where $1 \leq i \leq \lambda$, denote the progress of an offspring of a parent of fitness k in one step if this progress is positive; otherwise $Y_i = 0$ since only offspring of non-negative progress can be selected. Then, $Y_i \sim \max\{0, Y_i^1 - Y_i^0\}$, where $Y_i^0 \sim \mathrm{Bin}(n-k, c/n)$ and $Y_i^1 \sim \mathrm{Bin}(k, c/n)$ denote the random number of zeros and ones flipped by mutation, respectively. Clearly, $Y_i \succ Y_i^1 - Y_i^0$ and $\max\{Y_i \mid 1 \leq i \leq \lambda\} \succ \max\{Y_i^1 - Y_i^0 \mid 1 \leq i \leq \lambda\}$. From Lemma 6 it follows that

$$\Delta_t = \max\{Y_i \mid 1 \leq i \leq \lambda\} \succ \max\{Y_i^1 \mid 1 \leq i \leq \lambda\} - Y_1^0,$$

where the index 1 is without loss of generality and refers to an arbitrary of the λ offspring. Since $\max\{Y_i^1 \mid 1 \leq i \leq \lambda\} = X^*$, by linearity of expectation this yields $E(\Delta_t \,|\, X_t) \geq E(X^*) - c = (1-o(1))E(X^*)$ if $E(X^*) = \omega(1)$. Applying the third statement of Lemma 4 with $\alpha = (\ln \ln \ln \lambda)^{-1}$ recalling that $\lambda = \omega(1)$ provides us with the desired bound. Note this argument is specific for the first regime. For all the other regimes, we instead condition on the event that the offspring flipping most one-bits does not flip zero-bits, introducing the aforementioned factor of $(1-o(1))e^{-c}$ for each bound.

The second bound on the drift is obtained by applying the third statement of Lemma 4 with $\alpha = 1$.

The third bound on the drift stems from the second statement of Lemma 4 with $\alpha = \min\{c, 1\}$.

The fourth regime's bound on the drift stems from the second statement of Lemma 4 with $\alpha = c/\sqrt{\ln n}$. Finally, the last bound stems from the first statement of Lemma 4. Note also that h is monotone increasing w. r. t. X_t

(choosing the smallest $1-o(1)$ function from the five regimes and using $\lambda = \omega(1)$) and that its reciprocal is integrable on $\mathbb{R}^+$.

We have established bounds on the drift. Since the $(1+\lambda)$ EA clearly optimizes OneMax in expected finite time (as each generation has a positive probability of creating the optimum), we can apply Theorem 1 with minimal distance $x_{\min} = 1$ and get

$$E(T \mid X_0) \leq \frac{1}{h(1)} + \int_1^{X_0} \frac{1}{h(x)} \,\mathrm{d}x \ .$$

The first term is

$$\frac{1}{h(1)} = \frac{1}{(1-o(1))\lambda c e^{-c\frac{1}{n}}} = (1+o(1))\frac{e^c n}{\lambda c} \ ,$$

which is a lower order term in the total expected optimization time as we will see in the following.

We will now examine each regime's contribution to the expected optimization time by splitting up the integral at the corresponding bounds. Due to our definition of h the first four regimes' contribution can be easily computed since they do not depend on X_t.

For the first regime, we first determine X_0. Using a Chernoff bound, the probability that the initial individual is created within $[n/2 - n^{2/3}, n/2 + n^{2/3}]$ is at least $1 - 2e^{-\frac{2n^{1/3}}{3}}$, so we will condition on this in the following, introducing a small error probability absorbed by the $o(1)$-term of the theorem.

The first regime contributes a term of at most

$$\begin{aligned}
&\int_{\frac{n}{(\ln\lambda)\frac{1}{\ln\ln\ln\lambda}}}^{\frac{n}{2}+n^{\frac{2}{3}}} \frac{\mathrm{d}x}{(1-o(1))\frac{\ln\lambda}{\ln\ln\lambda}} \\
&\qquad \leq (1+o(1))\frac{\left(\frac{n}{2}+n^{\frac{2}{3}}\right)\ln\ln\lambda}{\ln\lambda} \\
&\qquad \leq (1+o(1))\frac{1}{2}\frac{n\ln\ln\lambda}{\ln\lambda} \ .
\end{aligned}$$

The second regime contributes a term of at most

$$\begin{aligned}
&\int_{\frac{n}{\ln\lambda}}^{\frac{n}{(\ln\lambda)\frac{1}{\ln\ln\ln\lambda}}} \frac{\mathrm{d}x}{(\frac{1}{2}-o(1))e^{-c}\frac{\ln\lambda}{\ln\ln\lambda}} \\
&\qquad \leq (2+o(1))\frac{e^c n\ln\ln\lambda}{(\ln\lambda)^{1+\frac{1}{\ln\ln\ln\lambda}}} \ .
\end{aligned}$$

This term is clearly asymptotically smaller than the first, due to the denominator's exponent.

The third regime contributes a term of at most

$$\int_{\frac{n}{\lambda}}^{\frac{n}{\ln\lambda}} \frac{\mathrm{d}x}{(1-o(1))e^{-c}\min\{c/2, 1/2\}} \leq \frac{(1+o(1))\,2e^c n}{\min\{c,1\}\ln\lambda} \ ,$$

and is thus asymptotically smaller than the first term as well.

The fourth regime contributes a term of at most

$$\int_{\frac{n}{\lambda\sqrt{\ln n}}}^{\frac{n}{\lambda}} \frac{\mathrm{d}x}{(1-o(1))e^{-c}\frac{c}{\sqrt{\ln n}}} \leq (1+o(1))\frac{e^c n\sqrt{\ln n}}{c\lambda} \enspace ,$$

which is dominated by the following term.

The fifth regime contributes a term of at most

$$\int_{1}^{\frac{n}{\lambda\sqrt{\ln n}}} \frac{e^c n}{(1-o(1))\lambda c x}\,\mathrm{d}x \leq (1+o(1))\frac{e^c n\ln n}{c\lambda} \enspace .$$

Summing up the individual contributions we end up with

$$E(T) \leq (1+o(1))\left(\frac{e^c}{c}\cdot\frac{n\ln n}{\lambda} + \frac{1}{2}\cdot\frac{n\ln\ln\lambda}{\ln\lambda}\right) \enspace ,$$

since the unconditional error introduced by the conditioning on the initialization is of lower order. Note that this holds for the case of $\lambda = \omega(1)$.

In case of $\lambda = O(1)$ the first four regimes can be subsumed; using the same arguments as in the previous case we have $E(\Delta_t \mid X_t) \geq \hat{h}(X_t)$, where

$$\hat{h}(X_t) := \begin{cases} (1-o(1))e^{-c}\frac{c}{\sqrt{\ln n}} & \text{if } X_t \geq \frac{n}{\lambda\sqrt{\ln n}} \\ (1-o(1))ce^{-c}\lambda X_t/n & \text{if } X_t < \frac{n}{\lambda\sqrt{\ln n}} \end{cases} \enspace .$$

The first regime contributes a term of at most

$$\int_{\frac{n}{\lambda\sqrt{\ln n}}}^{\frac{n}{2}+n^{\frac{2}{3}}} \frac{\mathrm{d}x}{(1-o(1))e^{-c}\frac{c}{\sqrt{\ln n}}} \leq \left(\frac{1}{2}+o(1)\right)\frac{e^c n\sqrt{\ln n}}{c} \enspace ,$$

which is clearly dominated by the second regime's contribution which is the same as in the previous case by reusing the same arguments. Hence, for $\lambda = O(1)$, we have $E(T) \leq (1+o(1))\left((e^c/c)n\ln n\right)$.

Combining both cases, i. e. for all $\lambda \leq n^r$, we end up with the desired bound of

$$E(T) \leq (1+o(1))\left(\frac{e^c}{c}\cdot\frac{n\ln n}{\lambda} + \frac{1}{2}\cdot\frac{n\ln\ln\lambda}{\ln\lambda}\right) \enspace .$$

Lower Bound. Using Theorem 2, we will now show the matching lower bound. Theorem 2 requires a monotone process and a function ξ bounding the progress towards the optimum. We define $\xi : \mathbb{R}^+ \to \mathbb{R}^+, x \mapsto x - \log_2 x - 1$.

Note that $\Pr(X_{t+1} < \xi(X_t)) \leq \Pr(X_{t+1} \leq \xi(X_t))$. Due to our assumption that $\lambda \leq n^r$ and using Lemma 3, we have

$$\begin{aligned} \Pr(X_{t+1} \leq \xi(X_t)) &\leq \lambda\binom{X_t}{\lceil\log_2 X_t + 1\rceil}\left(\frac{1}{n}\right)^{\log_2(X_t)+1} \\ &\leq n^r\left(\frac{eX_t}{n\lceil\log_2(X_t)+1\rceil}\right)^{\log_2(X_t)+1} \enspace . \end{aligned}$$

Setting $x_{\min} := 2^r$, the last term assumes its maximum at $X_t = x_{\min}$ within $[x_{\min}, \dots, n]$ for sufficiently large n and takes the value r'/n^{r+1}, where $r' = ((ex_{\min})/(r+1))^{r+1}$. We condition on the event that $X_{t+1} \geq \xi(X_t)$ holds for the at most $O(n \ln n)$ generations we consider; by a union bound this introduces an error of $o(1)$. Note also that this condition only decreases the drift.

We will now distinguish between the cases $\lambda < \ln^2 n$ and $\lambda \geq \ln^2 n$. In the latter case, λ is above the cut-off point. Then the bound on the drift will be relatively simple and use only a single expression, as detailed later. We first thoroughly analyze the case of $\lambda < \ln^2 n$, where we have to be more careful as this includes the cut-off point. We claim that $E(\Delta_t \mid X_t) \leq h^*(X_t)$, where

$$h^*(X_t) := \begin{cases} (1+o(1))\frac{\ln \lambda}{\ln \ln \lambda} & \text{if } X_t \geq \frac{n}{\lambda\sqrt{\ln n}} \\ (1+o(1))ce^{-c}\lambda X_t/n & \text{if } X_t < \frac{n}{\lambda\sqrt{\ln n}} \end{cases},$$

To show that $h^*(X_t)$ indeed is a bound on the drift, we look into the two regions. In the region $X_t \geq \frac{n}{\lambda\sqrt{\ln n}}$, we simply estimate the drift from above by the number of flipping one-bits. Using the third statement of Lemma 4 with $\alpha = 0$ and observing that the number of flipping one-bits is maximized at $X_t = n$, we obtain the claimed bound $E(\Delta_t \mid X_t) \leq (1+o(1))\frac{\ln \lambda}{\ln \ln \lambda}$.

In the region $X_t < \frac{n}{\lambda\sqrt{\ln n}}$, the function $h^*(X_t)$ contains the factor e^{-c}, which has to be explained carefully. Here we will distinguish between two cases. Let F be the event that at least one of the λ offspring flips at least two one-bits. By the law of total probability

$$E(\Delta_t \mid X_t) = E(\Delta_t \mid X_t; F) \cdot \Pr(F) + E\big(\Delta_t \mid X_t; \overline{F}\big) \cdot \Pr(\overline{F}).$$

Using our assumption on X_t and taking a union bound over λ offspring, we get $\Pr(F) \leq \lambda\binom{X_t}{2}(c/n)^2 \leq \lambda X_t(n/(\lambda\sqrt{\ln n}))(c/n)^2 = O(X_t/(n\sqrt{\ln n}))$. Furthermore, we have $E(\Delta_t \mid X_t; F) \leq 2 + \lambda X_t c/n = 2 + o(1)$. This holds since after 2 bits have been flipped, the remaining bits flip with probability at most c/n each. The total expected number of one-bits flipped in addition to the two bits, counted over all offspring, is at most $\lambda X_t c/n$. Hence,

$$E(\Delta_t \mid X_t; F) \cdot \Pr(F) = (2+o(1))O(X_t/(n\sqrt{\ln n})) = O(X_t/(n\sqrt{\ln n})).$$

We now bound the drift on the event $\overline{F}$. Let Y_i, where $1 \leq i \leq \lambda$, denote the progress (decrease of OneMax-value) of the i-th offspring of a parent of fitness X_t in one step, maximized with 0 since only offspring of non-negative progress can be selected. Then, $Y_i \sim \max\{0, Y_i^1 - Y_i^0\}$, where $Y_i^0 \sim \mathrm{Bin}(n - X_t, c/n)$ and $Y_i^1 \sim \mathrm{Bin}(X_t, c/n)$. On $\overline{F}$, we have $Y_i = 0$ for those individuals where $Y_i^0 > 0$. Let Z_i be the event that offspring i does not flip any zero-bits, i. e., $Y_i^0 = 0$. Hence, by the law of total probability, $Y_i = Y_i^1 \cdot \Pr(Z_i)$, and therefore,

$$E\big(\Delta_t \mid X_t; \overline{F}\big) \leq E\left(\max_{i=1,\dots,\lambda} Y_i^1 \mid X_t; \overline{F}\right) \cdot \Pr(Z_i).$$

Now,

$$\Pr(Z_i) = \left(1 - \frac{c}{n}\right)^{n - X_t} \leq e^{-(c/n)(n-o(n))} = (1+o(1))e^{-c}$$

by our assumption on X_t, and

$$E\left(\max_{i=1,\dots,\lambda} Y_i^1 \mid X_t; \overline{F}\right) \leq \lambda X_t c/n,$$

using the first statement of Lemma 4 with X_t many variables and noting that $\overline{F}$ only reduces the number of flipping one-bits. Taking everything together, we obtain

$$E(\Delta_t \mid X_t) = (1+o(1))e^{-c}\lambda X_t c/n + O(X_t/(n\sqrt{\ln n})) = (1+o(1))\lambda c e^{-c} X_t/n$$

in the region $X_t < \frac{n}{\lambda\sqrt{\ln n}}$. This completes the proof of the claim $E(\Delta_t \mid X_t) \leq h^*(X_t)$.

Note that h^* is increasing. Note also that ξ is strictly increasing on $[1/\ln 2, \infty)$, hence ξ^{-1} exists on the corresponding domain and it is increasing as well. More precisely, $\xi^{-1} : [(1+\ln\ln 2/\ln 2) - 1, \infty) \to [1/\ln 2, \infty)$. We can define $\tilde{h} := h^* \circ \xi^{-1}$. By applying Theorem 2 we obtain

$$\begin{aligned} E(T \mid X_0) &\geq \frac{x_{\min}}{\tilde{h}(x_{\min})} + \int_{x_{\min}}^{X_0} \frac{1}{\tilde{h}(u)}\,\mathrm{d}u \\ &= \frac{x_{\min}}{h^*(\xi^{-1}(x_{\min}))} + \int_{\xi^{-1}(x_{\min})}^{\xi^{-1}(X_0)} \frac{\left(1-\frac{1}{x}\right)}{h^*(x)}\,\mathrm{d}x \ . \end{aligned}$$

The last equality is due to integration by substitution and due to $(\xi^{-1})' = (\xi')^{-1}$ which holds for ξ on $[1/\ln 2, \infty)$. Hereinafter, we abbreviate $h^{**}(x) := h^*(x)/\left(1-\frac{1}{x}\right)$.

Due to the definition of $x_{\min}$ the first term is

$$\frac{2^r}{\tilde{h}(2^r)} = \frac{2^r}{(1+o(1))\lambda c e^{-c\frac{2\cdot 2^r}{n}}} = (1-o(1))\frac{e^c n}{2\lambda c} \ ,$$

where we used $x \leq 2(x-1) - \log_2(x) = \xi(2x)$ for $x \geq 4$ and thus $\xi^{-1}(x) \leq 2x$ for $x \geq 4$ due to the monotonicity of ξ^{-1}. This is a lower order term as we will see in the following.

Regarding the integral, we again split the integral into the regimes that we bounded the drift on, in order to analyze those individually. We get

$$\int_{\xi^{-1}(x_{\min})}^{\xi^{-1}(X_0)} \frac{\mathrm{d}x}{h^{**}(x)} = \int_{\xi^{-1}(x_{\min})}^{\frac{n}{\lambda\sqrt{\ln n}}} \frac{\mathrm{d}x}{h^{**}(x)} + \int_{\frac{n}{\lambda\sqrt{\ln n}}}^{\xi^{-1}(X_0)} \frac{\mathrm{d}x}{h^{**}(x)} \ .$$

We start with the first integral. We have

$$\begin{aligned}
\int_{\xi^{-1}(x_{\min})}^{\frac{n}{\lambda\sqrt{\ln n}}} \frac{1}{h^{**}(x)}\,\mathrm{d}x &\geq \int_{2x_{\min}}^{\frac{n}{\lambda\sqrt{\ln n}}} \frac{n\left(1-\frac{1}{x}\right)}{(1+o(1))ce^{-c}\lambda x}\,\mathrm{d}x \\
&= \frac{e^c n}{(1+o(1))c\lambda}\int_{2x_{\min}}^{\frac{n}{\lambda\sqrt{\ln n}}}\left(\frac{1}{x}-\frac{1}{x^2}\right)\mathrm{d}x \\
&= \frac{e^c n}{(1+o(1))c\lambda}\Bigg(\ln n - \ln\lambda - \frac{\ln\ln n}{2} \\
&\quad - \ln(2x_{\min}) + \frac{\lambda\sqrt{\ln n}}{n} - \frac{1}{2x_{\min}}\Bigg) \\
&\geq (1-o(1))\frac{e^c n\ln n}{c\lambda}
\end{aligned}$$

since $\lambda < \ln^2 n$.

Similar to the proof of the upper bound, we have that the algorithm initializes with high probability in $[n/2-n^{2/3}, n/2+n^{2/3}]$ by using a Chernoff bound, i. e., $X_0 \geq n/2 - n^{2/3}$. In the following we condition on this event, introducing only a small error absorbed by the $o(1)$-term.

Now, for the second integral it holds that

$$\begin{aligned}
\int_{\frac{n}{\lambda\sqrt{\ln n}}}^{\xi^{-1}(X_0)} \frac{1}{h^{**}(x)}\,\mathrm{d}x &\geq \int_{\frac{n}{\lambda\sqrt{\ln n}}}^{X_0} \frac{\ln\ln\lambda\left(1-\frac{1}{x}\right)}{(1+o(1))\ln\lambda}\,\mathrm{d}x \\
&\geq \frac{\ln\ln\lambda}{(1+o(1))\ln\lambda}\left(\frac{n}{2}-n^{\frac{2}{3}}-\frac{n}{\lambda\sqrt{\ln n}}\right) \\
&\quad - \frac{\ln\ln\lambda}{(1+o(1))\ln\lambda}\left(\ln\left(\frac{\frac{n}{2}-n^{\frac{2}{3}}}{\frac{n}{\lambda\sqrt{\ln n}}}\right)\right) \\
&\geq \frac{\ln\ln\lambda}{(1+o(1))\ln\lambda}\left(\frac{(1-o(1))n}{2}\right) \\
&\geq (1-o(1))\frac{1}{2}\frac{n\ln\ln\lambda}{\ln\lambda}\ ,
\end{aligned}$$

where the first inequality follows from $\xi^{-1}(x) \geq x$ and the third inequality is due to the subtracted summand being a lower order term.

If $\lambda \geq \ln^2 n$, we have that $E(\Delta_t \mid X_t) \leq \tilde{h}^*(X_t)$, where

$$\tilde{h}^*(X_t) := (1+o(1))\frac{\ln\lambda}{\ln\ln\lambda}$$

for all $X_t \geq 0$. Reusing the analysis of the contribution of the first region in $h^*(X_t)$ above, we get the lower bound

$$(1-o(1))\frac{1}{2}\frac{n\ln\ln\lambda}{\ln\lambda}$$

in this case, which is the leading term in the bound from the theorem since λ is above the cut-off point.

Now we can give a bound on the expected runtime for all $\lambda \leq n^r$. Summing up the individual contributions, we end up with

$$E(T) \geq (1 - o(1)) \left(\frac{e^c}{c} \cdot \frac{n \ln n}{\lambda} + \frac{1}{2} \cdot \frac{n \ln \ln \lambda}{\ln \lambda} \right) \ ,$$

since the unconditional error introduced by the conditioning is of lower order. □

4 Experiments

While the analysis of the (1+λ) EA is asymptotically tight, it is still asymptotic in nature. Since the bound given in Theorem 7 does not capture the effect of the lower-order term on the expected runtime, some phenomena might be obscured for practically relevant values of n.

Hence, we performed experiments in order to illustrate the effect of c on the runtime for small and moderate problem sizes. We implemented the (1+λ) EA in C using the GNU Scientific Library for the generation of pseudo-random numbers.

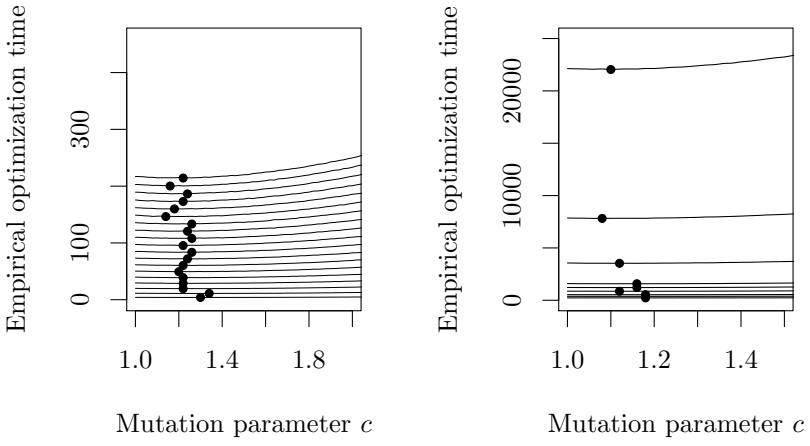

Figure 1: Empirical number of generations needed to optimize OneMax, averaged over 50000 runs of the $(1 + 5)$ EA for $n \in \{5, 10, \ldots, 95\}$ (left) and $n \in \{100, 150, 200, 300, 400, 500, 1000, 2000, 5000\}$ (right).

The results are displayed in Figure 1. We used an offspring population size of $\lambda = 5$ for all experiments. Both diagrams show the number of generations that the (1+5) EA with mutation probability c/n needed to optimize OneMax

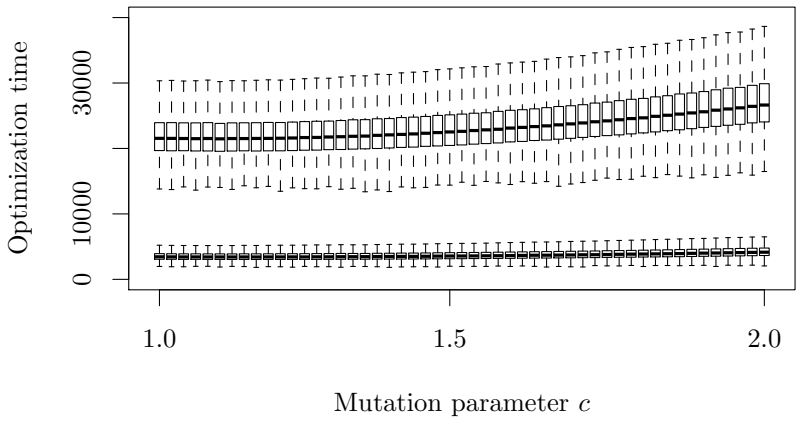

Figure 2: Number of generations needed to optimize OneMax as a box plot over 50000 runs for each value of c from 1.0 to 1.5 (step size 0.02) for $n = 1000$ (lower boxes) and $n = 5000$ (upper boxes). The outliers are not displayed.

as a function of the mutation parameter c. The optimization times are averaged over 50000 independent runs for problem size n, where $n \in \{5, 10, \ldots, 95\}$ in the left diagram (resp. $n \in \{100, 150, 200, 300, 400, 500, 1000, 2000, 5000\}$ in the right diagram). The parameter c takes values in the interval $[1.0, 2.0]$ (resp. $[1.0, 1.5]$ in the right diagram) with step size 0.02. For each n, the value for c that minimizes the empirical optimization time is marked.

The left digaram indicates that the optimal value for c varies around 1.3 for small problem sizes. This variation is due to the variance of the runs and the fact that for the considered values of n, similar optimization times are attained for a broad interval of c values, as can be seen in the plot. For example: for $n = 100$ the observed standard deviation of the optimization time is 67.62 for $c = 1$ (mean 231.54) and 162.97 for $c = 1.5$ (mean 412.98). For $n = 5000$ the observed standard deviation of the optimization time is 3482.51 for $c = 1$ (mean 22126.00) and 8555.14 for $c = 1.5$ (mean 45416.49). For illustration of the high variances observed the optimization times for $n = 1000$ and $n = 5000$ are displayed as a box plot in Figure 2.

It is not unexpected that the optimal mutation rate is higher than 1 for small values of n. This behaviour has already been observed for the (1+1) EA on OneMax for small problem sizes [CSWA15]. Furthermore, we can see that the slopes around the optimum value of c get steeper for higher values n. This can be explained by the leading constant e^c/c of the expected runtime, which grows exponentially in c around the optimal value of c and appears more pronounced

for larger values of n.

The empirical optimal c-values are small for all problem sizes; even for $n = 5000$ the empirical optimal value for c is still 1.1, which is only slightly smaller than 1.3 and subject to variance as well. However, the plot indicates that c approaches 1 for higher problem sizes, as predicted by our bound in Theorem 7.

Conclusions

We have presented the first tight runtime analysis of a population-based EA, depending on population size and mutation probability. More precisely, we have analyzed the well-known (1+λ) EA with mutation probability c/n, where $c > 0$ is a constant, on the classical OneMax function. Our results show that $1/n$ is the only optimal mutation probability if $\lambda = o(\ln n \ln \ln n / \ln \ln \ln n)$, which is the cut-off point for linear speed-up. However, if λ is above this cut-off point then the standard mutation probability $1/n$ is no longer the only optimal choice and the algorithm is more robust with respect to this choice. In fact, the expected number of generations is independent of c then (up to lower order terms), irrespectively of c being less than 1 or greater.

Our results shed light on the interplay of population size and mutation probability in evolutionary algorithms. At the same time, we have extended our reservoir of methods for the analysis. We are optimistic that our study paves the ground for tight analyses of more complex population-based EAs on also more complex function classes.

Acknowledgements

This work was supported by the Danish Council for Independent Research (DFF), grant no. 4002–00542. The authors thank the anonymous reviewers for their useful comments which helped to improve this work.

References

[AD11] Anne Auger and Benjamin Doerr, editors. *Theory of Randomized Search Heuristics: Foundations and Recent Developments*. World Scientific Publishing, 2011.

[BDN10] Süntje Böttcher, Benjamin Doerr, and Frank Neumann. Optimal fixed and adaptive mutation rates for the leadingones problem. In *Proc. of Parallel Problem Solving from Nature (PPSN 2010)*, volume 6238, pages 1–10. Springer, 2010.

[BLS14] Golnaz Badkobeh, Per Kristian Lehre, and Dirk Sudholt. Unbiased black-box complexity of parallel search. In *Proc. of Parallel Problem Solving from Nature (PPSN 2014)*, volume 8672 of *Lecture Notes in Computer Science*, page 892–901, 2014.

[CSWA15] Francisco Chicano, Andrew M. Sutton, L. Darrell Whitley, and Enrique Alba. Fitness probability distribution of bit-flip mutation. *Evolutionary Computation*, 23(2):217–248, 2015.

[DFW10] Benjamin Doerr, Mahmoud Fouz, and Carsten Witt. Quasirandom evolutionary algorithms. In *Proc. of the Genetic and Evolutionary Computation Conference (GECCO 2010)*, page 1457–1464. ACM Press, 2010.

[DFW11] Benjamin Doerr, Mahmoud Fouz, and Carsten Witt. Sharp bounds by probability-generating functions and variable drift. In *Proc. of the Genetic and Evolutionary Computation Conference (GECCO 2011)*, pages 2083–2090. ACM Press, 2011.

[DG13] Benjamin Doerr and Leslie Ann Goldberg. Adaptive drift analysis. *Algorithmica*, 65(1):224–250, 2013.

[DJW12] Benjamin Doerr, Daniel Johannsen, and Carola Winzen. Multiplicative drift analysis. *Algorithmica*, 64(4):673–697, 2012.

[DK13] Benjamin Doerr and Marvin Künnemann. Royal road functions and the (1+λ) evolutionary algorithm: Almost no speed-up from larger offspring populations. In *Proc. of the IEEE Congress on Evolutionary Computation (CEC 2013)*, pages 424–431. IEEE Press, 2013.

[DK15] Benjamin Doerr and Marvin Künnemann. Optimizing linear functions with the (1+λ) evolutionary algorithm – different asymptotic runtimes for different instances. *Theoretical Computer Science*, 561:3–23, 2015.

[GW15] Christian Gießen and Carsten Witt. Population size vs. mutation strength for the (1+λ) EA on OneMax. In *Proc. of Genetic and Evolutionary Computation Conference (GECCO 2015)*, pages 1439–1446. ACM Press, 2015.

[Jäg11] Jens Jägersküpper. Combining markov-chain analysis and drift analysis – the (1+1) evolutionary algorithm on linear functions reloaded. *Algorithmica*, 59(3):409–424, 2011. Preliminary version in Proc. of PPSN '08.

[Jan13] Thomas Jansen. *Analyzing Evolutionary Algorithms - The Computer Science Perspective*. Natural Computing Series. Springer, 2013.

[JJW05] Thomas Jansen, Kenneth A. De Jong, and Ingo Wegener. On the choice of the offspring population size in evolutionary algorithms. *Evolutionary Computation*, 13(4):413–440, 2005.

[Joh10] Daniel Johannsen. *Random combinatorial structures and randomized search heuristics*. PhD thesis, Universität des Saarlandes, Germany, 2010.

[LW14] Per Kristian Lehre and Carsten Witt. Concentrated hitting times of randomized search heuristics with variable drift. In *Proc. of ISAAC '14*, volume 8889 of *Lecture Notes in Computer Science*, pages 686–697. Springer, 2014. Full technical report at `http://arxiv.org/abs/1307.2559`.

[MRC09] Boris Mitavskiy, Jonathan E. Rowe, and Chris Cannings. Theoretical analysis of local search strategies to optimize network communication subject to preserving the total number of links. *International Journal of Intelligent Computing and Cybernetics*, 2(2):243–284, 2009.

[NW10] Frank Neumann and Carsten Witt. *Bioinspired Computation in Combinatorial Optimization – Algorithms and Their Computational Complexity*. Natural Computing Series. Springer, 2010.

[RS14] Jonathan E. Rowe and Dirk Sudholt. The choice of the offspring population size in the $(1, \lambda)$ evolutionary algorithm. *Theoretical Computer Science*, 545:20–38, 2014. Preliminary version in Proc. of GECCO 2012.

[Sud12] Dirk Sudholt. Crossover speeds up building-block assembly. In *Proc. of the Genetic and Evolutionary Computation Conference (GECCO 2012)*, pages 689–702. ACM Press, 2012.

[Sud13] Dirk Sudholt. A new method for lower bounds on the running time of evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 17(3):418–435, 2013. Preliminary version in Proc. of PPSN '10.

[Wit06] Carsten Witt. Runtime analysis of the $(\mu + 1)$ EA on simple Pseudo-Boolean functions. *Evolutionary Computation*, 14(1):65–86, 2006.

[Wit13] Carsten Witt. Tight bounds on the optimization time of a randomized search heuristic on linear functions. *Combinatorics, Probability & Computing*, 22(2):294–318, 2013. Preliminary version in Proc. of STACS '12.